Sun GlassFish Communications Server 1.5 Release Notes

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820–4276–10
January 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun^{TM} Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc., ou ses filiales, aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

090122@21808

◆ ◆ ◆ CHAPTER 1

Overview

Sun GlassFish™ Communications Server adds capabilities of Session Initiation Protocol (SIP) Servlet technology to Sun GlassFish Enterprise Server.

Sun GlassFish Communications Server is free for development, deployment, and redistribution. If you are a customer who is interested in redistribution, contact Sun OEM sales for a redistribution license.

Sun GlassFish Communications Server is an easy, fast, and industry-leading application server based on Java™ Platform, Enterprise Edition (Java EE) and SIP technology for developing and delivering web applications and web services. Sun GlassFish Communications Server enables you to build SIP applications and converged applications that are JSR 289–complaint. It provides superior performance, clustering, and high availability features for scalable, carrier-grade services that continue to operate despite software and hardware faults.

- "About These Notes" on page 3
- "Accessibility Features" on page 4
- "Related Documentation" on page 4
- "How to Report Problems and Provide Feedback" on page 5
- "Sun Welcomes Your Comments" on page 6
- "Additional Sun Resources" on page 6

About These Notes

These Release Notes contain important information available at the time of release of Sun GlassFish Communications Server 1.5. Enhancements, known problems, and other late-breaking issues are addressed here. Read this document before you begin using Communications Server 1.5.

The most up-to-date version of these release notes can be found at the Communications Server documentation web site (`http://docs.sun.com/coll/1343.8`). Check the web site prior to installing and setting up your software and then periodically thereafter to view the most up-to-date release notes and product documentation.

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Accessibility Features

To obtain accessibility features that have been released since the publishing of this media, consult Section 508 product assessments available from Sun upon request to determine which versions are best suited for deploying accessible solutions. Updated versions of applications can be found at: `http://sun.com/software/javaenterprisesystem/get.html`

For information on Sun's commitment to accessibility, visit `http://sun.com/access`.

Related Documentation

TABLE 1–1 Books in the Communications Server Documentation Set

Book Title	Description
Documentation Center	Communications Server documentation topics organized by task and subject.
Release Notes	Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, Java Development Kit (JDK™), and database drivers.
Quick Start Guide	How to get started with the Communications Server product.
Installation Guide	Installing the software and its components.
Application Deployment Guide	Deployment of applications and application components to the Communications Server. Includes information about deployment descriptors.
Developer's Guide	Creating and implementing Java Platform, Enterprise Edition (Java EE platform) applications intended to run on the Communications Server that follow the open Java standards model for Java EE components and APIs. Includes information about developer tools, security, debugging, and creating lifecycle modules.
Java EE 5 Tutorial	Using Java EE 5 platform technologies and APIs to develop Java EE applications.
Java WSIT Tutorial	Developing web applications using the Web Service Interoperability Technologies (WSIT). Describes how, when, and why to use the WSIT technologies and the features and options that each technology supports.

TABLE 1–1 Books in the Communications Server Documentation Set *(Continued)*

Book Title	Description
Administration Guide	System administration for the Communications Server, including configuration, monitoring, security, resource management, and web services management.
High Availability Administration Guide	Setting up clusters, working with node agents, and using load balancers.
Administration Reference	Editing the Communications Server configuration file, `domain.xml`.
Performance Tuning Guide	Tuning the Communications Server to improve performance.
Reference Manual	Utility commands available with the Communications Server; written in man page style. Includes the `asadmin` command line interface.

How to Report Problems and Provide Feedback

If you have problems with Sun GlassFish Communications Server, contact Sun using one of the following mechanisms:

- SailFin mailing lists (`https://sailfin.dev.java.net/servlets/ProjectMailingListList`) — A variety of SailFin community mailing lists for various interests and feedback
- Bug database on Java Developer Connection (`http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml`) — To view bugs or to submit a bug, use the Java Developer Connection Bug Parade
- SailFin Issue Tracker (`https://sailfin.dev.java.net/servlets/ProjectIssues`)
- Java Technology Forums (`http://forum.java.sun.com/`) — An interactive message board for sharing knowledge and questions about Java technologies and programming techniques.
- Sun Software Support services (`http://www.sun.com/service/sunone/software`) — Links to the Knowledge Base, Online Support Center, and Product Tracker, as well as to maintenance programs and support contact numbers
- The telephone dispatch number associated with your maintenance contract

So that we can best assist you in resolving problems, please have the following information available when you contact support:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to `http://docs.sun.com` and click Feedback. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the title of this book is *Sun GlassFishCommunications Server 1.5 Release Notes*, and the part number is 820-4276.

Additional Sun Resources

Useful information can be found at the following locations:

- Communications Server product information (`http://wwws.sun.com/software/products/appsrvr/home_appsrvr.html`)
- Professional Services (`http://www.sun.com/service/sunps/sunone`)
- Software Products and Service (`http://www.sun.com/software`)
- Software Support Services (`http://www.sun.com/service/sunone/software`)
- Support and Knowledge Base (`http://www.sun.com/service/support/software`)
- Sun Support and Training Services (`http://training.sun.com`)
- Consulting and Professional Services (`http://www.sun.com/service/sunps/sunone`)
- Developer Information (`http://developers.sun.com`)
- Sun Developer Support Services (`http://www.sun.com/developers/support`)
- Software Training (`http://www.sun.com/software/training`)
- Sun Software Data Sheets (`http://wwws.sun.com/software`)
- Sun Microsystems product documentation (`http://docs.sun.com/`)

◆ ◆ ◆ CHAPTER 2

About Sun GlassFish Communications Server 1.5

Communications Server is a Java EE 5 platform-compatible server for the development and deployment of Web, SIP, and converged applications and Java technology-based web services in large-scale production environments.

This chapter includes:

- "Features of Sun GlassFish Communications Server " on page 7
- "Hardware and Software Requirements" on page 10
- "Using the Update Center" on page 17

Features of Sun GlassFish Communications Server

While Communications Server supports all the features available in Sun GlassFish Enterprise Server, it has a host of new features as listed here. For the list of features in Sun GlassFish Enterprise Server, see the Sun GlassFish Enterprise Server Release Notes.

- **Compliant with JSR 289** — Communications Server contains a SIP stack that is compliant with RFCs 3261, 3262, 3265, 3311, 3515, 3903, etc. It will also expose the SIP Servlets Java APIs defined by JSR 289.
- **Converged SIP and HTTP Sessions** — Communications Server supports converged HTTP and SIP session functionality as defined by JSR 289.
- **Integrated SIP Servlet Container** — The SIP Servlet container offers the following functionality:
  - Provides a network end point to listen to SIP requests,
  - Provides an environment to host and manage the lifecycle of SIP Servlets.

  The SIP Servlet container works in conjunction with the web container to provide support for converged applications as defined by the SIP Servlet API Specification (SSA).
- **Easy Administration** — Communications Server enables easy administration of the SIP container and the deployed SIP applications using the CLI and the GUI.

- **Security**— Digest Authentication is supported for both HTTP and SIP protocols. P-Asserted-Identity is supported for SIP protocol. The SIP Container supports SIPS over TLS.
- **Converged Load Balancer** — Converged Load Balancer (CLB) is a software load balancing component, which facilitates high availability of converged applications, by distributing the application requests in a cluster of server instances. The CLB provides the following functionality:
  - Forwards SIP(S) and HTTP(S) requests to a cluster of server instances.
  - Load balances requests using consistent hashing/round-robin policy.
  - Enables and disables server instances in a cluster.
  - Monitors health of server instances.
- **Support to Deploy SIP and Converged Applications** — Communications Server supports deployment of pure SIP applications and converged applications using the extension-module mechanism in GlassFish. It also supports `sun-sip.xml` as the Sun-specific deployment descriptor for SIP applications.
- **NetBeans Tooling** — Communications Server provides support for SIP Servlet applications based on JSR 289 on top of NetBeans 6.0. A simple SIP client can be created using the test agent to test SIP Servlets. This plug-in is available under `as-install-dir/lib/tools`.
- **Application Router Deployment** — Communications Server supports hot deployment of Application router archives (JAR files) using the extension module mechanism. An application router that follows a simple alphabetical rule for routing the requests is provided by default.
- **Packaging and File Layout** — Communications Server is being delivered as a file-based installable.
- **Registration with Sun Connection** — You can use the installer, the Admin Console GUI, or the Update Center to register the product with Sun Connection (`http://www.sun.com/service/sunconnection/index.jsp`). By registering the Communications Server with Sun Connection you receive benefits such as:
  - Patch information and bug updates
  - Screencasts and tutorials
  - News and events
  - Support and training offerings
- **SuSe Linux 64–bit support**
- **Multilevel relationship prefetching support**— Multilevel relationship prefetching for container-managed persistence (CMP) 2.1 entity beans is now supported. For more information, see "Enabling Multilevel Relationship Prefetching" on page 18.
- **Enhanced JBI support** — You can update a JBI component through the Admin Console GUI or from the command line without the need to redeploy any service assemblies that are already deployed.

- **Java EE 5 platform support** — Sun GlassFish Communications Server implements the Java EE 5 specification to deliver one of the best application runtimes for next-generation enterprise applications and web services. Communications Server implements the following Java EE standards:
  - Enterprise Java Beans 3.0
  - JAXB 2.0
  - Java Persistence
  - Java Server Faces 1.2
  - Java Server Pages 2.1 (JSP 2.1)
  - Java Server Pages Standard Tag Library (JSTL) 1.2
  - Streaming API for XML (StAX)
  - Web Services Metadata
  - Java API for XML based Web Services 2.0 (JAX-WS 2.0)
  - Common Annotations for the Java Platform 1.0 (CAJ 1.0)
  - Java Servlet 2.5

  The complete list of Java EE 5 platform technologies is provided later in these notes.
- **Web Services Interoperability Technologies (WSIT) support** – Sun is working closely with Microsoft to ensure interoperability of Web services enterprise technologies such as message optimization, reliable messaging, and security. The initial release of WSIT is a product of this joint effort. WSIT is an implementation of a number of open web services specifications to support enterprise features. In addition to message optimization, reliable messaging, and security, WSIT includes a bootstrapping and configuration technology. Starting with the core XML support currently built into the Java platform, WSIT uses or extends existing features and adds new support for interoperable web services, including:
  - Bootstrapping and Configuration
  - Message Optimization Technology
  - Reliable Messaging Technology
  - Security Technology

  See "More About WSIT Integration" on page 19 later in this chapter for more information about WSIT integration in Communications Server.
- **JBI support** – JBI extends Java EE with business integration Service Provider Interfaces (SPI). These SPI enable developers to create or implement a Java business integration environment for specifications such as WSCI, BPEL4WS and the W3C Choreography Working Group. A JBI implementation is installed directly by the Communications Server installer, mostly in the *as-install*`/jbi` directory. This directory contains all common JAR files and system components for the JBI including a `lifecycle` module that starts the JBI framework in the Communications Server JVM.
- **Usage profiles** – Every administrative domain is associated with a usage profile, which identifies the capabilities of that domain. Communications Server provides the following profiles:

- *Developer* – Use this profile if you are running your domain in a development environment and if your applications do not need clustering features, such as load balancing, high availability, and session replication. Note that the actual name of the profile is "`developer`" (case sensitive).
- *Cluster* – Use this profile if you want to create clusters of application server instances imparting scalability and high availability to the deployed Java EE applications. The state of the applications is persisted *in-memory*. Note that the actual name of the profile is "`cluster`" (case sensitive).

- **Open Source and SailFin Community** — TheSailFin (`http://sailfin.dev.java.net`) community has the goal of developing a free, open source, commerical-grade communcations server that implements the newest features of the Java EE 5 platform and related enterprise technologies.
- **Update Center Support** – The Communications Server Update Center provides automated Communications Server updates and easy access to additional components.

Hardware and Software Requirements

This section lists the requirements that must be met before installing Sun GlassFish Communications Server.

- "Supported Platforms" on page 10
- "Important Patch Information" on page 11
- "JDK Version" on page 11

Supported Platforms

The following table lists the operating systems with which the Sun GlassFish Communications Server is compatible.

TABLE 2–1 Supported Operating Systems

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM	Deployment Type
Sun Solaris 10 (x86)(32–bit and 64–bit)	512 MB	512 MB	250 MB free	500 MB free	J2SE 5.0 - 32 bit Java SE 6 - 32 bit	Production
Sun Solaris 10 (SPARC)(32–bit and 64–bit)	512 MB	512 MB	250 MB free	500 MB free	J2SE 5.0 - 32 bit Java SE 6 - 32 bit	Production

TABLE 2–1 Supported Operating Systems *(Continued)*

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM	Deployment Type
Red Hat Enterprise Linux 3.0, 4.0 (32-bit and 64-bit)	512 MB	1 GB	250 MB free	500 MB free	J2SE 5.0 - 32 bit Java SE 6 - 32 bit	Production
SuSE Linux Enterprise Server 10 SP1(32–bit and 64–bit)	512 MB	1 GB	250 MB free	500 MB free	J2SE 5.0 - 32 bit Java SE 6 - 32 bit	Production
Windows Vista	1 GB	2 GB	500 MB free	1 GB free	J2SE 5.0 - 32 bit Java SE 6 - 32 bit	Development
Mac OS	512 MB	1 GB	250 MB free	500 MB free	J2SE 5.0 - 32 bit Java SE 6 - 32 bit	Development

On UNIX™, you can check your operating system version using the `uname` command. Disk space can be checked using the `df` command.

Note – Its recommended that you use the NTFS file system rather than FAT or FAT32 when running the Communications Server on any Microsoft Windows platform.

Important Patch Information

Solaris Patch Requirements

It is recommended that Solaris 10 (x86, SPARC) users have the "Sun recommended patch cluster" installed. This patch cluster is available under "Recommended and Security Patches" on the SunSolve (`http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage`) Web site.

JDK Version

The minimum (and certified) version of JDK required for Communications Server is **1.6.0_7**.

▼ To Switch to the Supported Java SE Version

You can switch to a supported Java SE version by editing the `asenv file` as described here.

1. **If you have not already done so, install the new Java SE version on your system.**
   The Java SE SDK can be downloaded from `http://java.sun.com/javase`
2. **Stop the Communications Server domains.**
3. **Edit the *as-install*`/config/asenv.conf` file (`asenv.bat` on Windows), changing the value for `AS_JAVA` to point to the new Java home directory:**
4. **Edit the *as-install*`/samples/common.properties` file, changing the line beginning `com.sun.aas.javaRoot...` to reference the new Java home directory.**
5. **Restart Communications Server.**

Ant Version

The Ant version bundled with SailFin JAR installation is **1.6.5**. To build a custom Application Router application, you need Ant **1.7**, which you will need to download separately.

JDBC Drivers and Databases

Table 2–2 lists databases and drivers that meet the Java EE compatibility requirements. All supported configurations of the Sun GlassFish Communications Server must contain at least one database/driver combination from this table, such as the bundled Java DB database and driver. In addition, the Communications Server is designed to support JDBC connectivity to any additional DBMS with a corresponding JDBC driver.

TABLE 2–2 Java EE-Compatible JDBC Drivers

JDBC Driver Vendor	JDBC Driver Type	Supported Database Server
Derby Network Client	Type 4	Derby 10.2
DataDirect 3.6.*x*, 3.7.*x*	Type 4	Oracle 10g Oracle 9i Sybase ASE 15 MS SQL 2005 DB2 9.1 DB2 8.2
MySQL Connector/J Driver 5.1.6	Type 4	MySQL 5.0

TABLE 2–2 Java EE-Compatible JDBC Drivers *(Continued)*

JDBC Driver Vendor	JDBC Driver Type	Supported Database Server
Oracle 10G	Type 4	Oracle 10g -RAC
PostGres	Type 4	8.2.*x*

Table 2–3 lists all the additional databases and drivers scheduled to be supported in the Communications Server FCS release.

TABLE 2–3 Supported JDBC Drivers and Databases (Communications Server FCS)

JDBC Driver Vendor	JDBC Driver Type	Supported Database Server
Oracle OCI	Type 2	Oracle 9i
IBM DB2	Type 2	DB2 9.1

In general, the Communications Server supports all JDBC drivers that meet the Java EE specification.

Message Queue Versions

For file-based installations, Message Queue 4.3 is co-packaged with Communications Server. For UNIX package-based installations, Message Queue 4.1.1, which is functionally equivalent to 4.3, is provided. Despite the version numbering, Message Queue 4.1.1 is actually backward-compatible with Message Queue 4.2.

The embedded Sun GlassFish Message Queue code that is supplied as part of Communications Server is only tested and certified (typically) against the equivalent version of the Message Queue broker. This means that using the supplied embedded Message Queue code to connect to a remote (not managed by Communications Server) Message Queue broker running a different version of the Message Queue code is not supported.

Browsers

This section lists the browsers that are supported with the Sun GlassFish Communications Server 2.1.

TABLE 2–4 Supported Web Browsers

Browser	Version
Mozilla	1.7.12
Internet Explorer	6.0 Service Pack 2, 7.0

TABLE 2–4 Supported Web Browsers *(Continued)*

Browser	Version
Firefox	1.5.*x*, 2.*x*
Safari	2
Netscape	8.0.4, 8.1, 9.0, 9.0.*x*

Browsers

This section lists the browsers that are supported with the Sun GlassFish Communications Server 2.1.

TABLE 2–5 Supported Web Browsers

Browser	Version
Mozilla	1.7.12
Internet Explorer	6.0 Service Pack 2, 7.0
Firefox	1.5.*x*, 2.*x*
Safari	2
Netscape	8.0.4, 8.1, 9.0, 9.0.*x*

Other Requirements

The following additional requirements should be met before installing the Sun GlassFish Communications Server software.

- **Free space:** your temporary directory must have a minimum of 35MB free for Sun GlassFish Communications Server installation, and 250 MB of free space for the SDK installation.
- **Using the uninstall program:** If you need to remove the Communications Server from your system, it is important to use the uninstall program that is included with the software. If you attempt to use another method, problems will arise when you try to reinstall the same version, or when you install a new version.
- **Free ports:** You must have seven unused ports available. The installation program automatically detects ports in use and suggests currently unused ports for the default domain. By default, the initial default ports are:
  - `8080` for HTTP, `8181` for HTTPS
  - `3700` for IIOP, `3820` for IIOP/SSL and `3920` for IIOP/SSL with Mutual Authentication
  - `4848` (HTTP) for the Administration Server
  - `8686` (RMI) port for pure JMX Clients

- **Shutting down firewall** (Microsoft Windows) — You must stop any firewall software before installing the Sun GlassFish Communications Server software, because some of this software disables all ports by default. The installation program must be able to accurately determine which ports are available.

Java EE 5 Platform APIs

The Sun GlassFish Communications Server 2.1 supports the Java EE 5 platform. The following table lists the enhanced APIs available on the Java EE 5 platform.

TABLE 2–6 Major API changes on the Java EE 5 Platform

API	JSR
Java EE 5	
Java Platform, Enterprise Edition 5 (http://java.sun.com/javaee/5/docs/api/)	JSR 244 (http://jcp.org/aboutJava/communityprocess/pr/jsr244/)
Web Services Technologies	
Implementing Enterprise Web Services	JSR 109 (http://jcp.org/en/jsr/detail?id=109)
Java API for XML-Based Web Services (JAX-WS) 2.0 (https://jax-ws.dev.java.net/)	JSR 224 (http://jcp.org/en/jsr/detail?id=224)
Java API for XML-Based RPC (JAX-RPC) 1.1 (https://jax-rpc.dev.java.net/)	JSR 101 (http://jcp.org/en/jsr/detail?id=101)
Java Architecture for XML Binding (JAXB) 2.0 (https://jaxb.dev.java.net/)	JSR 222 (http://jcp.org/en/jsr/detail?id=222)
SOAP with Attachments API for Java (SAAJ) (https://saaj.dev.java.net/)	JSR 67 (http://jcp.org/en/jsr/detail?id=67)
Streaming API for XML (http://java.sun.com/webservices/docs/1.6/tutorial/doc/SJSXP.html)	JSR 173 (http://jcp.org/en/jsr/detail?id=173)
Web Service Metadata for the Java Platform	JSR 181 (http://jcp.org/en/jsr/detail?id=181)
Component Model Technologies	
Enterprise JavaBeans 3.0 (http://java.sun.com/products/ejb/)	JSR 220 (http://jcp.org/en/jsr/detail?id=220)
J2EE Connector Architecture 1.5 (http://java.sun.com/j2ee/connector/)	JSR 112 (http://jcp.org/en/jsr/detail?id=112)
Java Servlet 2.5 (http://java.sun.com/products/servlet/)	JSR 154 (http://jcp.org/en/jsr/detail?id=154)

TABLE 2–6 Major API changes on the Java EE 5 Platform *(Continued)*

API	JSR
JavaServer Faces 1.2 (http://java.sun.com/j2ee/javaserverfaces/)	JSR 252 (http://jcp.org/en/jsr/detail?id=252)
JavaServer Pages 2.1 (http://java.sun.com/products/jsp/)	JSR 245 (http://jcp.org/en/jsr/detail?id=245)
JavaServer Pages Standard Tag Library 1.2 (http://java.sun.com/products/jsp/jstl/)	JSR 52 (http://jcp.org/en/jsr/detail?id=52)
Management Technologies	
J2EE Management (http://java.sun.com/j2ee/tools/management/)	JSR 77 (http://jcp.org/en/jsr/detail?id=77)
J2EE Application Deployment (http://java.sun.com/j2ee/tools/deployment/)	JSR 88 (http://jcp.org/en/jsr/detail?id=88)
Java Authorization Contract for Containers (http://java.sun.com/j2ee/javaacc/)	JSR 115 (http://jcp.org/en/jsr/detail?id=115)
Other Java EE Technologies	
Common Annotations for the Java Platform	JSR 250 (http://jcp.org/en/jsr/detail?id=250)
Java Transaction API (JTA) (http://java.sun.com/products/jta/)	JSR 907 (http://jcp.org/en/jsr/detail?id=907)
JavaBeans Activation Framework (JAF) 1.1 (http://java.sun.com/products/javabeans/glasgow/jaf.html)	JSR 925 (http://jcp.org/en/jsr/detail?id=925)
JavaMail (http://java.sun.com/products/javamail/)	JSR 919 (http://jcp.org/en/jsr/detail?id=919)
Java Message Service API (http://java.sun.com/products/jms/)	JSR 914 (http://www.jcp.org/en/jsr/detail?id=914)
Java Persistence API (http://java.sun.com/j2ee/persistence/faq.html)	JSR 220 (http://www.jcp.org/en/jsr/detail?id=220)

Java EE 5 SDK

Sun GlassFish Communications Server 2.1 is available as part of the Java EE 5 SDK.

There are two Java EE 5 SDK versions:

- Java Application Platform SDK (http://java.sun.com/javaee/downloads/)
- Java EE 5 SDK (http://java.sun.com/javaee/downloads/index.jsp)

In addition, you can download these SDK distributions with the JDK or with the NetBeans **5.5.1** tools. For more information, access the download page at `http://java.sun.com/javaee/downloads/index.jsp`.

Note – GlassFish v2.1 and Communications Server 2.1 *does not work* with NetBeans 5.5 for developing Java EE applications. You *must* use NetBeans **5.5.1**.

Switching to Another Supported Java Version

Sun GlassFish Communications Server 2.1 requires Java SE 5.0 or greater as the underlying JVM. If you want to switch from one Java version to another, perform the following general steps. (Windows and Unix)

Note – Downgrading to an earlier Java version is not recommended after a domain has been created with a newer Java VM. If you must downgrade your JVM, it is recommended that you do it on a per-domain basis. The following procedure describes how to do this.

▼ To Switch to Another Supported Java Version

1. **Download the desired Java SDK (not the JRE) and install it on your system, if you have not already done so.**

   The Java SDK can be downloaded from `http://java.sun.com/j2se`.

2. **Start the domain for which you want to change the JDK:**

   as-install`/bin/asadmin start-domain` *domain-name*

3. **Log in to the Admin console and change the JVM attributes for the domain.**

   In particular, you may want to change the JAVA_HOME variable on the JVM Settings page for the domain.

   Alternatively, you can use the `asadmin` command:

   as-install`/bin/asadmin set "server.java-config.java-home=`*Path-To-Java-Home*"

Using the Update Center

The Update Center provides automated Communications Server updates and easy access to additional components.

When the Update Center is enabled, it performs an automated software update. During this automated update process, the Update Center collects and transmits the following data to Sun Microsystems (or its service provider):

- Unique installation ID (GUID)
- IP address
- Operating system information (name, version, architecture, locale)
- JDK version
- Module download information (module name, date, time, status, download time, number of bytes downloaded)

No personally identifiable information is tracked. No personally identifiable information is associated with any other data or used for reporting purposes.

▼ To Use the Update Center

To ensure explicit agreement of the automated update, the Update Center is disabled by default. To enable the Update Center to perform periodic checks and automated updates:

1. **Start the Update Center.**
   - On Unix: `<installdir>/updatecenter/bin/updatetool`
   - On Windows: `<installdir>\updatecenter\bin\updatetool.bat`
2. **Select the Preferences tab.**
3. **In the Update Scheduling window, change the Check for Updates drop-down box value from Never (Manual) to a desired value. For example, daily or weekly.**
4. **Specify the desired day of the week and time of the day for the update.**
5. **Select the Save button to save your changes.**

   The Update Center will now automatically check for SailFin or Communications Server component updates according to the schedule specified. When an update is available, the Update Center will launch and notify you of the component available to update.

Enabling Multilevel Relationship Prefetching

Multilevel relationship prefetching is supported for CMP 2.1 entity beans.

To enable multilevel relationship prefetching, set the following property:

```
-Dcom.sun.jdo.spi.persistence.support.sqlstore.MULTILEVEL_PREFETCH=true
```

For more information about relationship prefetching, see "Relationship Prefetching" in *Sun GlassFish Communications Server Developer's Guide*.

More About WSIT Integration

For detailed information about WSIT status, refer to the WSIT Status Notes (https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes/status-notes-1-0-1-FCS.html) page.

◆ ◆ ◆ CHAPTER 3

Communications Server Known Issues and Limitations

This chapter describes known problems and associated solutions for the Communications Server software. If a summary statement does not specify a particular platform, the problem applies to all platforms. This information is organized into the following sections:

- "Communications Server Administration" on page 21
- "Converged Load Balancer" on page 22
- "Installation" on page 23
- "Security" on page 23
- "SIP Container" on page 24

Communications Server Administration

Communications Server instances start even if SIP/SIPS ports are not bound (Issue Number 998)

Description

Communications Server instances starts up even if it cannot bind to a SIP or SIPS port.

Solution

Ensure that ports are free before starting server instance(s). Check the log files (server.log) to ensure that there have not been any SIP container errors or exceptions during the startup.

For details on how to view log files, see TBDlink.

Communications Server does not use the JDK specified by —`javahome` option (Issue Number 789)

Description

You can use a preinstalled JDK instead of the default version for the installation by using the —`javahome` option. Communications Server, by default, uses the JDK version from *as-install*`/jdk`.

Solution

The `AS_JAVA` variable in the `asenv.conf` file always points to *as-install*`/jdk`. If you want to use a different JDK version, update the `asenv.conf` file manually and changed the value of `AS_JAVA`.

Using 3.5 GB Java heap causes instances to restart while traffic is on (Issue 1169)

Description

When JVM heap size is set to 3.5 GB, Communications Server instances fail and restart when they receive traffic.

Solution

Ensure that the maximum JVM heap size is set to 3.0 GB or less. For details on how to change the JVM heap size, see TBDlink

Communications Server wrongly reports CPU usage when using only one of the cores of a multi-core system (Issue 1344)

Description

On Solaris platforms, Communications Server calculates CPU usage based on the number of available processors and the per-core CPU usage. However, Communications Server takes into account the static value of number of cores and not the number of cores that are used by the JVM.

Solution

Recalculate the CPU threshold values if you are not using all the cores in the machine.

Converged Load Balancer

Using converged load balancer asadmin commands changes domain status to "requires restart" (Issue 333)

Description

When you use `asadmin` commands related to converged load balancer, the domain status changes to "Restart required", although it is not necessary to restart the domain. You can view the domain status with the `asadmin list-domains` command.

Solution

Domain restart is not required. Ignore the domain status.

SEVERE messages in logs due to dynamic reconfiguration of converged load balancer after application deployment (Issue 1161)

Description

If you modify the configuration of the converged load balancer on a target and you redeploy the applications on that target, the instance logs will show SEVERE messages.

Solution

These messages do not affect the functioning of the converged load balancer or the instances. Ignore these messages.

When complete URI is used, the BEKey parameter in the Contact header is not correctly escaped (Issue 1466)

Description

When you use a converged load balancer with a data-centric rules file that returns a complete URI for the BEKey parameter, he BEKey parameter in the Contact header is not correctly escaped. The ":" character is not correctly escaped as specified in RFC 3261.

Solution

No known solution.

Installation

Communications Server installer crashes on Linux (6739013)

Description

This problem has been observed on systems running Linux with the environment variable, `MALLOC_CHECK_`, set to 2.

Solution

Set the environment variable, `MALLOC_CHECK_` to 0. Run the export command as follows:

```
export MALLOC_CHECK_
```

Security

Communications Server throws exception when trust-auth-realm-ref property is not specified in sun-sip.xml (CR 6786131)

Description

Communications Server throws Null pointer exception Realm is not configured when P-Asserted-Identity authentication is configured in `sun-sip.xml`.

Solution

Configure Realm using the `trust-auth-realm-ref` property in `sun-sip.xml`. See TBDlink for more information on setting this property.

SIP Container

SIP container unable to handle a CANCEL when it has sent a 100 response (Issue 712)

Description

The SIP container cannot handle a CANCEL request when a 100 response has been sent.

Solution

The application needs to send a provisional response (such as 1xx), so that the remote side is able to CANCEL the INVITE request.

SIP sessions and HTTP sessions do not apply the same session expiration time model (Issue 1180)

Description

The session expiration model of SIP sessions is different from the HTTP expiration time logic. In HTTP, the session is automatically extended, outside of the control of the application, whenever a new HTTP request is received in that HTTP session.

With SIP sessions, the application is in control over the duration of the SipApplicationSession (SAS), subject to approval by the SIP container. Applications can use the `setExpires` method to indicate when the SAS should expire. `setExpires` defines an expiration time relative to the moment the `setExpires` method is called. The container can modify, reject, or accept the duration indicated in `setExpires`. If the session is not invalidated, then the `sessionExpired` callback is performed at the time defined by `setExpires` . In this callback, the application can try to extend the duration of the SAS by invoking a new `setExpires`, again subject to modification, rejection, or acceptance by the container.

For this reason, converged applications that start out with the same expiration time of the SipApplicationSession (SAS) and on the HTTP session will notice that the SAS times out before the HTTP session if new requests were received on the HTTP session.

Solution

The best way to deal with the different expiration time handling of the SIP and the HTTP session is to start with a large enough SAS expiration time, which is the total time that the application session is expected to live (including several HTTP requests). The SAS lifetime could even be set to infinite, specifically if `invalidateWhenReady` semantics are used, in which case, the SipApplicationSession is invalidated when the last protocol child session becomes invalidated. The initial expiration time for the SAS can be configured in the deployment descriptor.

If the maximum total duration can be estimated in advance, no further code is needed, as it is then appropriate to invalidate both the Sip session and the HTTP session when the SAS expires. If the maximum duration cannot be estimated in advance, then the SipApplicationSession can be extended when it expires, as shown in the code snippet below.

In the `SipApplicationSessionListener` implementation, you can do something like this:

```
public void sessionExpired(SipApplicationSessionEvent sasEvent) {
                // check if the SAS needs to be extended first, if so:
        int granted = sasEvent.getApplicationSession().setExpires(2);
        if (granted <= 0) {
            System.out.println("extension rejected");
        } else if (granted < 2) {
            System.out.println("extension granted with lower value " + granted);
        } // else allowed
    }
```

SIP session lives on after container callback to sessionExpired (Issue 1265)

Description

This is an intermittent issue. The SIP container intermittently responds with a 500 Server internal error message instead of a 481 Call/Transaction does not exist message when there is a race condition between the 200 for NOTIFY indicating that the session has been removed, and the SUBSCRIBE sent by the client when receiving that NOTIFY.

Solution

The client needs to the refresh SUBSCRIBE much before the subscription expires.

Communications Server first acts as a UAS, then as a proxy, and generates NOP (Issue 1432)

Description

When it receives an INVITE request, Communications Server first acts as UAS, replies to this request with 1XX, and then proxies this INVITE request to another instance, which replies with 200 OK. The 1xx creates an internal virtual branch while the 200 message creates a real branch. Upon reception of 200 OK from B the internal virtual branch should be cancelled

Solution

This exception trace does not affect the functionality of the virtual proxy branch.

getLastAccessedTime method does not provide accurate results (Issue 1351)

Description

The `getLastAccessedTime` method of a SIP session does not provide accurate results.

Solution

Applications that need to keep accurate track of the `lastAccessedTime` must store it themselves into the `SipApplicationSession`.

```
synchronized (sas) {
    Long last = (Long) sas.getAttribute("myLastAccessedTime");
    if (last == null) {last = 0};
    // do something with the last one
    // and...
    // set the new one.
    sas.setAttribute("myLastAccessedTime", System.currentTimeMillis());
}
```

SIP listener remains active for a certain duration after it is deleted (Issue 1294)

Description

After a SIP listener configured for TCP and UDP requests is deleted, the listener remains active for a certain duration. UDP requests that are sent to the listener could receive a response from the listener.

Solution

No known solution. The SIP listener stops listening to UDP requests after a certain duration. This problem does not impact TCP requests.

Communications Server throws exception when it receives a Contact header without "<>" (Issue 1489)

Description

Communications Server throws exception when it receives a Contact header without "<>". According to SIP RFC 3261, it is not mandatory to have the "<>" in the address. This could lead to interoperability problems with other SIP-compliant devices.

Solution

Use "<>" in the Contact header.

Communications Server throws exception at an invalid UUID value (Issue 1494)

Description

Communications Server throws exception at an invalid UUID value instead of returning a 400 Bad Request. The UUID value resides in the SIP contact header's `sip.instance` value

Solution

No known solution.

Windows: Sometimes, UDP messages are not received by Communications Server (No id)

Description

This problem is intermittently seen only on Windows. UDP messages are not received by Communications Server.

Solution

Set the following JVM option as follows, and restart Communications Server.

```
org.jvnet.glassfish.comms.disableUDPSourcePort=true
```

◆ ◆ ◆ CHAPTER 4

Sun GlassFish Enterprise Server Known Issues and Limitations

This chapter describes known problems and associated workarounds for the Sun GlassFish Enterprise Server 2.1 software. These problems specifically also affect Communications Server, which is built on top of Sun GlassFish Enterprise Server 2.1. If a summary statement does not specify a particular platform, the problem applies to all platforms. This information is organized into the following sections:

- "Administration" on page 29
- "Application Client" on page 34
- "Bundled Sun JDBC Drivers" on page 35
- "Deployment" on page 36
- "Documentation" on page 36
- "Installation" on page 37
- "Java EE Tutorial" on page 39
- "Java Persistence" on page 39
- "Lifecycle Management" on page 40
- "Logging" on page 41
- "Message Queue" on page 41
- "Monitoring" on page 42
- "Security" on page 43
- "Web Container" on page 45

Administration

This section describes known administration issues and associated solutions.

Failed to Deploy Generic RA Resource Adapter against IBM MQ (Issue 6605)

Description

Deployment of a generic RA adapter against IBM Message Queue product fails. The permissions granted in the `server.policy` file is as follows.

```
grant {
      permission java.util.logging.LoggingPermission "control";
      permission java.util.PropertyPermission "*", "read,write";
      }
```

Solution

Change the permissions in the `server.policy` file as follows:

```
grant codeBase
"file:${com.sun.aas.installRoot}/lib/install/applications/adminapp/-" {
  permission java.util.logging.LoggingPermission "control";
 };
```

The `package-appclient` script does not work if `domain1` is not present (6171458)

Description

By default, there is a hard-coded value in *as-install*`/lib/package-appclient.xml` for the `AS_ACC_CONFIG` variable for `domain1` that is pointed to by `asenv.conf`. If `domain1` is deleted and a new domain created, the `AS_ACC_CONFIG` variable is not updated with the new domain name, which causes the `package-appclient` script to fail.

Solution

Do one of the following:

1. Leave `domain1` intact, and create your other domains around it.
2. Remove `domain1` and replace the hard-coded value for `domain1` in *as-install*`/lib/package-appclient.xml` with the new domain name.

This will have to be done every time a new domain is created if `domain1` is not present.

Starting Application Server with additional JMX Agent is not supported (6200011)

Description

J2SE 1.4.x, 5.0, or later can be configured on the Application Server. An integral feature of J2SE 5.0 platform is the ability to start a JMX agent. This is activated when you explicitly set system properties at the server startup.

Example values include:

```
name="com.sun.management.jmxremote" value="true"
name="com.sun.management.jmxremote.port" value="9999"
name="com.sun.management.jmxremote.authenticate" value="false"
name="com.sun.management.jmxremote.ssl" value="false"
```

After configuring JMX properties and starting the server, a new `jmx-connector` server is started within the Application Server Virtual Machine. An undesirable side-effect of this is that the administration functions are affected adversely, and the Application Server administration Console and command—line interface may produce unexpected results. The problem is that there are some conflicts between the built in `jmx-connector` server and the new `jmx-connector` server.

Solution

If using `jconsole` (or any other `JMX-compliant` client), consider reusing the standard JMX Connector Server that is started with Application Server startup.

When the server starts up, a line similar to the one shown below appears in the server.log. You can connect to the `JMXService` URL specified there and perform the same management/configuration operations after successfully providing the credentials; for example:

```
[#|2004-11-24T17:49:08.203-0800|INFO|sun-appserver-ee8.1|
javax.enterprise.system.tools.admin|_ThreadID=10;|ADM1501:
Here is the JMXServiceURL for the JMXConnectorServer:
[service:jmx:rmi:///jndi/rmi://hostname:8686/management/
rmi-jmx-connector]. This is where the remote administrative
clients should connect using the JSR 160 JMX Connectors.|#]
```

For more information, refer to the *Sun GlassFish Communications Server 1.5 Administration Guide*.

`.asadmintruststore` file not described in the documentation (6315957)

Description

The `.asadmintruststore` file is not described in the documentation. If this file does not exist in the server administrator's `home` directory, you may experience serious bugs when upgrading certain applications hosted on the server.

Solution

- If possible, the `asadmin start-domain domain1` command should be run by user who installed the server.
- If it is not run by that user, the `.asadmintruststore` should be moved or copied from the `home` directory of installing user to the `home` directory of the running user.
- Note that if the file is moved (not copied) from the installing user's `home` directory to the running user's `home` directory, you might experience application upgrade problems, as described in bugs 6309079, 6310428 and 6312869, because the upgrade/install user (normally `root` in Java ES) will no longer have the `.asadminstruststore` file in his or her home directory.

Clustered instances fail to start due to a timeout in reaching the JMS broker (6523663)

Description

The default MQ integration mode for a Communications Server cluster instance is `LOCAL`. When Communications Server is installed in a location (`PATH`) that is long (read "`not short`"), `imqbrokerscv.exe` crashes when the cluster instance starts. The problem is a memory allocation problem in `imqbrokersvc`.

Solution

The JMS service type for the cluster instance must be changed from the default `LOCAL` to `REMOTE`. In this configuration, all the instances point back to the DAS broker. Follow the instructions below to configure a cluster in `REMOTE` mode.

Note – When using `REMOTE` mode, all instances are using one broker (DAS) , and therefore no broker cluster is created when the Communications Server cluster starts up. See "Auto-clustering" in Section 4.1, Division iii of the one-pager at `http://www.glassfishwiki.org/gfwiki/attach/OnePagersOrFunctionalSpecs/as-mq-integration-gfv2.txt` for more information. The above functionality will not be available!

▼ Using the command-line

Before You Begin Modify the port and password file according to your environment. Note that in the instructions below, the cluster name is `racluster`, the DAS admin port is `5858`, and the DAS JMS port is `7676`.

1 Modify the cluster configuration, changing the JMS type to `REMOTE`.

```
as-install/bin/asadmin.bat set --port 5858 --user admin --passwordfile \
as-install/bin/password_file racluster.jms-service.type=REMOTE
```

2 Create a JMS host corresponding to the DAS JMS host.

```
as-install/bin/asadmin.bat create-jms-host --port 5858 --user admin --passwordfile \
as-install/bin/password_file --target racluster --mqhost localhost --mqport 7676 \
--mquser admin --mqpassword admin dashost
```

3 Set the default JMS host to be the DAS JMS host created in the previous step.

```
as-install/bin/asadmin.bat set --port 5858 --user admin --passwordfile \
as-install/bin/password_file racluster.jms-service.default-jms-host=dashost
```

Using the Admin GUI

1. **Go to Configurations->*cluster-name*-config->Java Message Service->JMS Hosts.**
2. **Click *New* to create a new JMS host; name it `dashost`.**
3. **Enter configuration settings corresponding to the JMS service for the DAS; defaults are as follows:**
   - Hostname: `localhost`
   - Port: `7676`
   - Admin user: `admin`
   - Password: `admin`

   Modify these settings as appropriate for your DAS JMS service.
4. **Navigate back to the Java Message Service tab, and change the JMS service type to `REMOTE` (default is `LOCAL`).**
5. **Choose `dashost` from the `default-jms-host` drop-down list.**
6. **Save the changes, and then start your node-agent or cluster.**

Cannot display jmaki chart in Netscape 8.1.3, Mozilla 1.7 and Safari 2.0.4 browsers (6543014)

Description

When trying to display a chart from the Log Statistics Monitoring page using some unsupported browsers, the following error may be thrown:

```
Error loading jmaki.widgets.jmaki.charting.line.Widget : id=form1:jmaki_chart11
Script: http://easqelx5.red.iplanet.com:4848/resources/jmaki/charting/ \
line/component.js (line:5437).
Message: area.initialize is not a function
```

Solution

Use a supported browser. Refer to "Browsers" on page 14 for a list of browsers supported by Communications Server.

The `create-domain` command fails with custom master password in AIX (6628170)

Description

On the AIX operating system, an attempt to create a domain with a custom master password fails with the following error:

```
keytool error (likely untranslated): java.lang.NullPointerException
 Enter keystore password:  New keystore password:
```

Application Client

This section describes known application client issues and associated solutions.

Library JAR packaged in Application Client Archive overwrites MANIFEST file (6193556)

Description

If you have a top level JAR file inside your client JAR (in this case, reporter.jar), when you deploy the client JAR, the MANIFEST file for that JAR overwrites the MANIFEST file for the client JAR.

Solution

None at this time.

ACC always tries to connect to `localhost:3700` (6527987)

Description

The application client always tries to connect to `localhost:3700`. The problem is that several system properties need to be read before the client code is invoked.

Solution

Set the following as system properties (`-D` in your `JAVA_CMD`). Do *not* set them in your appclient code:

```
org.omg.CORBA.ORBInitialHost = server-instance-host
org.omg.CORBA.ORBInitialPort = server-instance-port
```

`ASQuickStartup` breaks `SocketChannel.keyFor()`, returning null instead of the `SelectionKey` (Issue Tracker 3027)

Description

When a `SocketChannel` is registered on several Selectors, doing `socketChannel.keyFor(lastRegisteredSelector)` returns null instead of the `SelectionKey`.

Solution

This is related to a JDK bug, 6562829, and is expected to be fixed in 6.0 U3. A workaround has been included in Communications Server, such that the selector is unwrapped before the `keyFor` API is called. This enables the `keyFor` to succeed until JDK bug is fixed.

Bundled Sun JDBC Drivers

This section describes known bundled Sun JDBC driver issues and associated solutions.

`PreparedStatement` errors (6170432)

Description 1

If an application generates more than `3000` `PreparedStatement` objects in one transaction, the following error may occur with DB2:

```
[sunm][DB2 JDBC Driver] No more available statements.Please recreate your
package with a larger dynamicSections value.
```

Solution 1

Add following properties to the connection pool definition to get the driver to rebind DB2 packages with a larger dynamic sections value:

`createDefaultPackage=true replacePackage=true dynamicSections=1000`

See the *Sun GlassFish Communications Server 1.5 Administration Guide*for details about configuring connection pools.

Description 2

Related to the `PrepardStatement` error above, another error message that may be thrown is:

```
[sunm][DB2 JDBC Driver][DB2]Virtual storage or database resource is not
available.
```

Solution 2

Increase the DB2 server configuration parameter *APPLHEAPSZ*. A good value is `4096`.

Description 3

Isolation level `TRANSACTION_SERIALIZABLE`. If your application uses isolation level `TRANSACTION_SERIALIZABLE` and uses one of the parameters suggested above, it might hang while obtaining a connection.

Solution 3

To set desired isolation level for a connection, the corresponding connection pool has to be created at that isolation level. See the *Sun GlassFish Communications Server 1.5 Administration Guide* for instructions.

Java DB is not started after machine reboot or Application Server start (6515124)

Description

The bundled Java DB database is not automatically restarted after a host system or Solaris zone reboot, or an Communications Server start. This is not a bug, but expected behavior for any bundled or third-party application. The problem is that the Java DB must be started before the Communications Server instance.

Solution

After rebooting the host machine or Solaris zone, be sure to start the Java DB *before* starting Communications Server; for example:

```
/opt/SUNWappserver/appserver/bin/asadmin start-database
```

Refer to "Administration Tools" in *Sun GlassFish Communications Server 1.5 Quick Start Guide* in the *Sun GlassFish Communications Server 1.5 Quick Start Guide* for more information about `asadmin` command options.

Deployment

Autodeployment fails on a cluster sometimes (6610527)

Description

Timing issues sometimes cause autodeployment to fail in domains that are configured to support clusters. The issue is not observed in domains that do not support clusters.

Solution

Use one of the following solutions:

- Use autodeployment as follows:
  - Automatically deploy individual applications sequentially.
  - Introduce a delay between autodeployments of individual applications
- Deploy applications manually by using either the Admin Console GUI or the command line.

Documentation

This section describes known documentation issues and associated solutions.

Javadoc Inconsistencies (various IDs)

The Javadoc for several AMX interfaces and methods is either missing or incorrect:

- Getter methods for `NumConnAcquired` and `NumConnReleased` statistics are missing from `ConnectorConnectionPoolStats` and `AltJDBCConnectionPoolStats`. These getter methods will be added in a future release as `getNumConnAcquired()` and `getNumConnReleased()`.
- Calling the following methods in `EJBCacheStats` will throw an exception: `getPassivationSuccesses()`, `getExpiredSessionsRemoved()`, `getPassivationErrors()`, `getPassivations()`. This will be fixed in a future release.
- The AMX MBeans may require several seconds after server startup before they are all registered and available for use. A future release will make it possible to determine when the AMX MBeans are fully loaded.
- The constant `XTypes.CONNNECTOR_CONNECTION_POOL_MONITOR` is misspelled ("`NNN`"). This will be corrected in a future release.

Bundled ANT throws `java.lang.NoClassDefFoundError` (6265624)

Description

The following exception is thrown in thread "main" `java.lang.NoClassDefFoundError: org/apache/tools/ant/launch/Launcher`.

Solution

Use the bundled ANT for things outside the Communications Server is not recommended.

Installation

This section describes known installation issues and associated solutions.

Installation shutdown hanging on some Linux systems after clicking the "Finish" button (5009728)

Description

This problem has been observed on several Linux systems. It is most common on Java Desktop System 2 but has also been observed on Linux Red Hat distributions.

After clicking the "Finish" button on the last installer screen, the installer fails to launch a browser window containing the product About page or product registration page, and hangs indefinitely, not returning the command prompt.

Solution

Exit the installer by pressing Ctrl+C in the terminal window in which the installer was started. After doing this, browser window containing product About page or registration page will sometimes be launched, but if it does not show up, start the browser and enter following URL in order to review About page:

`file://`*as-install*`/docs-ee/about.html`

If you also selected the installation option to register the product, follow the link to registration page available on product About page.

On Windows, the `imq` directory needs to be created during installation (6199697)

Description

On Windows, immediately after installing Application Server Enterprise Edition, the Message Queue broker fails on startup with a message saying the directory `drive:\as\domains\domain1\imq` does not exist.

Note that if the broker is started after starting `domain1`, the directory will be created by the Application Server and the problem will not occur.

Solution

1. Create the `var_home_dir_location` before creating the broker:

   ```
   $imqbrokerd -varhome var_home_dir_location
   ```

   For example:

   ```
   $imqbrokerd -varhome D:\as\domains\domain1\imq
   ```

Uninstall does not update `productregistry` file properly; unable to use silent mode to install (6571598)

Description

If the Communications Server `productregistry` file contains shared component configurations, an Communications Server uninstallation procedure does not update the `productregistry` file correctly, and you will not be able to use silent mode in a subsequent installation unless the `productregistry` file is renamed or removed. Leaving the shared components entries in the `productregistry` file intact is by design, but it leads to confusion with subsequent silent installs.

Solution

After a successful uninstallation is reported back through uninstall log files, delete the `productregistry` file prior to running a subsequent installation. To verify that a previous uninstallation has completed successfully, look for a `appserv_uninstall.class` file in *as-install*. This file will *not* be present if the uninstallation was successful.

Note – Do not delete the `productregistry` if the uninstallation was not successful.

The `productregistry` file is located in `/var/sadm/install` on Solaris and `/var/tmp` on Linux.

Java EE Tutorial

When using the Administration Console to create any resource, use the Targets tab to specify the server as the target. If you use the command line or an `asant` target, the server is the default target, no further action is required.

Java Persistence

TopLink expects my `Collection` field/property to be cloneable (Issue Tracker 556)

Description

If the `java.util.Arrays.asList()` API is used to convert an `Object[]` to `Collection`, the JDK returns an implementation of `java.util.ArrayList` that is not cloneable. This results in the following exception:

```
The method invocation of the method [protected native java.lang.Object
java.lang.Object.clone() throws java.lang.CloneNotSupportedException] on the object
[[pkg.A id = xxx]], of class [class java.util.Arrays$ArrayList], triggered an
exception. Internal Exception: java.lang.reflect.InvocationTargetException Target
Invocation Exception: java.lang.CloneNotSupportedException:
java.util.Arrays$ArrayList
```

This issue is tracked at `https://glassfish.dev.java.net/issues/show_bug.cgi?id=556`.

Solution

Create another collection using its constructor; for example:

```
myCollection = new ArrayList(java.util.Arrays.asList(a))
```

`GenerationType.IDENTITY` and DataDirect Driver with SyBase (Issue Tracker 2431)

Description

An attempt to insert an entity that uses `GenerationType.IDENTITY` fails when the DataDirect driver is used with SyBase. The attempt fails because the DataDirect driver creates a stored procedure for every parameterized prepared statement.

Solution

In the `domain.xml` file, set the property `PrepareMethod=direct` on the corresponding data source.

Lifecycle Management

This section describes known lifecycle management issues and associated solutions.

Setting `ejb-timer-service` property causes `set` command to fail (6193449)

Description

After setting the `ejb-timer-service` property `minimum-delivery-interval` to `9000`, an attempt to set the `ejb-timer-service` property `redelivery-interval-in-mills` to `7000` causes the `set` command to fail with the following error:

```
[echo] Doing admin task set
[exec] [Attribute(id=redelivery-interval-internal-in-millis) :
Redelivery-Interval (7,000)
should be greater than or equal to Minimum-delivery-interval-
in-millis (9,000)]
[exec] CLI137 Command set failed.
```

- `minimum-delivery-interval` is the minimal interval duration between deliveries of the same periodic timer.
- `redelivery-interval-in-mills` is the time the timer service will wait after a failed `ejbTimeout` before attempting redelivery.

The problem is that the logic that relates the redelivery interval property to the minimum delivery property is incorrect and prevents you from using the GUI or the CLI to set any value where the minimum delivery interval is greater than redelivery interval.

The `minimum-delivery-interval-in-millis` must always be set equal to or higher than `ejb-timer-service` property `redelivery-interval-in-millis`. The problem is that there is an erroneous validation check in the Application Server to verify that the value for `redelivery-interval-in-millis` is greater than the value for `minimum-delivery-interval-in-millis`.

Solution

Use the default values for these properties, as follows:

```
minimum-delivery-interval(default)=7000
redelivery-interval-in-millis(default)=5000
```

Values other than these defaults will generate an error.

Error thrown when list JMS physical destinations within non-DAS config (6532532)

Description

If you are trying to view the JMS Physical Destinations using the default-config, you will see an error message.

Solution

This is expected behavior. In Communications Server 2.1, `default-config` is a template of configuration information and hence JMS operations (such as `list` and `create`) cannot be executed for the `default-config`. These JMS operations can, however, be executed for the configurations of your cluster or standalone instances.

Win2003 only: Non-paged pool leak memory, breaking tcp stack and richaccess test (6575349)

Description

(Windows 2003 only) There are memory leaks on Windows 2003 systems when performing rich access functions. The problem occurs because the Win32 nonpaged pool keeps growing, eventually bringing down the entire TCP/IP stack. Once the failure happens, the TCP/IP stack is left in an recoverable state, and the only way restore it is by rebooting the Windows 2003 system.

Workaround

There are two workarounds to this issue:

- Use Grizzly blocking mode by configuring the `domain.xml` `http-listener` attribute, `blocking-enabled="true"` or add the following `http-listener` property:

  ```
  <property name="blocking" value="true"/>
  ```

- Use Windows Vista or Windows XP.

Logging

This section describes known logging issues and solutions.

Setting debug statement for `access,failure` causes hang in Application Server startup (6180095)

Description

Setting the `java.security.debug` option for the JVM will cause the server instance startup to freeze with a deadlock; for example, setting the following in `domain.xml` causes the problem:

```
<jvm-options>-Djava.security.debug=access,failure</jvm-options>
```

Solution

None at this time. Please avoid setting this flag.

Message Queue

This section describes known Java message queue issues and associated solutions.

JMS reconnection does not successfully complete in certain cases that are timing dependent (6173308, 6189645, 6198481, 6199510, 6208728)

Description

Failures to reconnect in timing-dependent scenarios can be caused by several problems.

Solution

You can work around these problems by:

- Restarting the brokers involved
- Restarting the Application Server instances involved

MQ broker fails to start with cluster profile on Linux (6524871)

Description

After creating a domain with a cluster profile on a Linux system, you may encounter a `java.lang.OutOfMemoryError: Java heap space` error, and the server instance may fail to restart because the MQ broker does not start. The system never recovers after this condition. The problem is a misconfigured `/etc/hosts` file; specifically, the server host name is pointing to the loopback address `127.0.0.1`.

Solution

By design, an MQ broker cluster cannot start with the network device configured to point to the loopback address. This is not a bug. The solution is to make sure that the `/etc/hosts` file for the Communications Server host does not point to `127.0.0.1`.

Mismatch of old and new classes is created when imqjmsra.jar is loaded before upgrade (6740794)

Description

During Application Server startup, the server checks the Message Queue version. If the Message Queue version is incorrect, then the server upgrades using the `imqjmsra.jar`. This upgrade JAR and its classes will not be available to the server until the next restart of Application Server. This situation only occurs if Message Queue is upgraded alone, or if Application Server is patched alone. A side effect of this situation is that sometimes Application Server does not start.

Solution

Both Message Queue and Application Server need to be maintained at the same patch level, or restart the Application Server.

Monitoring

This section describes known monitoring issues and associated solutions.

Some of the HTTP Service monitoring statistics do not present useful information and should be ignored (6174518)

Description

When viewing the monitoring statistics of some elements of the HTTP Service, some values presented do not correspond to current values or are always 0. Specifically, the following HTTP Service statistics do not present information applicable to the Application Server, and should be ignored:

- `http-service`

  `load1MinuteAverage`

  `load5MinuteAverage`

  `load15MinuteAverage`

  `rateBytesTransmitted`

  `rateBytesReceived`
- `pwc-thread-pool` (the element)

Solution

These monitors will be removed in future releases and replaced with more appropriate information.

Open JNDI Browsing from Admin UI dumps a huge amount of exceptions in the `server.log` (6591734)

Description

Many exceptions are thrown when the JNDI browser is opened from the Admin GUI.

Solution

None at this time.

Security

This section describes known issues and associated solutions related to Communications Server and web application security and certificates.

OutofMemory Error in SSL Scenarios During Heavy Stress (JDK 6 Issue 23)

Description

A JDK bug (See: https://jdk6.dev.java.net/issues/show_bug.cgi?id=23) in JDK6 Sun PKCS11 Provider could cause an OutOfMemoryError when running certain SSL scenarios under heavy stress.

Solution

If you run into this issue, remove `sun.security.pkcs11.SunPKCS11` provider from the `java.security` file in your JRE installation.

SSL termination is not working (6269102)

Description

SSL termination is not working; when Load Balancer (Hardware) is configured for SSL termination, the Communications Server changes the protocol from `https` to `http` during redirection.

Solution

Add a software load balancer between the hardware load balancer and the Communications Server.

Socket connection leak with SSL (6492477)

Description

Because of a JVM bug, there is a leak issue with some JDK versions when `security-enabled` is set to `true` on an HTTP listener. Specifically, the steps to reproduce this bug are as follows:

1. Set `security-enabled` to `true` on the HTTP listener:

   ```
   <http-listener acceptor-threads="1" address="0.0.0.0"
   blocking-enabled="false" default-virtual-server="server" enabled="true"
   family="inet" id=" http-listener-1" port="8080" security-enabled="true"
   server-name="" xpowered-by="true">
   ```

2. Comment out stopping domain at the end of quicklook tests.
3. Run quicklook tests.
4. Check socket usage:

   ```
   netstat -an | grep 8080
   ```

   The following are shown to be in use:

   ```
   *.8080               *.*                0      0 49152      0 LISTEN
   *.8080               *.*                0      0 49152      0 BOUND
   ```

This issue is tracked on the GlassFish site at `https://glassfish.dev.java.net/issues/show_bug.cgi?id=849`.

Solution

Upgrade to the latest JDK version.

Web Container

This section describes known web container issues and associated solutions.

On Windows, deploying an application using `--precompilejsp=true` can lock JAR files in the application, causing later undeployment or redeployment to fail (5004315)

Description

If you request precompilation of JSPs when you deploy an application on Windows, later attempts to undeploy that application or to redeploy it (or any application with the same module ID) will not work as expected. The problem is that JSP precompilation opens JAR files in your application but does not close them, and Windows prevents the undeployment from deleting those files or the redeployment from overwriting them.

Note that undeployment succeeds to a point, in that the application is logically removed from the Application Server. Also note that no error message is returned by the `asadmin` utility, but the application's directory and the locked jar files remain on the server. The server's log file will contain messages describing the failure to delete the files and the application's directory.

Attempts to redeploy the application after undeploying fail because the server tries to remove the existing files and directory, and these attempts also fail. This can happen if you try to deploy any application that uses the same module ID as the originally deployed application, because the server uses the module ID in choosing a directory name to hold the application's files.

Attempts to redeploy the application without undeploying it first will fail for the same reasons.

Diagnostics

If you attempt to redeploy the application or deploy it after undeploying it, the asadmin utility returns an error similar to the one below.

```
An exception occurred while running the command. The exception
message is: CLI171 Command deploy failed : Deploying application in
domain failed; Cannot deploy. Module directory is locked and can’t
be deleted.
```

Solution

If you specify `--precompilejsps=false` (the default setting) when you deploy an application, then this problem will not occur. Be aware that the first use of the application will trigger the JSP compilation, so the response time to the first request will be longer than for later requests.

Note also that if you do precompile, you should stop and restart the server before undeploying or redeploying the application. The shutdown frees the locked JAR files so the undeployment or redeployment after the restart can succeed.

Unable to deploy WAR with Servlet 2.4-based `web.xml` that contains an empty `<load-on-startup>` element (6172006)

Description

The optional load-on-startup servlet element in a `web.xml` indicates that the associated servlet is to be loaded and initialized as part of the startup of the web application that declares it.

The optional content of this element is an integer indicating the order in which the servlet is to be loaded and initialized with respect to the web application's other servlets. An empty `<load-on-startup>` indicates that the order is irrelevant, as long as the servlet is loaded and initialized during the startup of its containing web application.

The Servlet 2.4 schema for `web.xml` no longer supports an empty `<load-on-startup>`, meaning that an integer must be specified when using a Servlet 2.4 based `web.xml`. If specifying an empty `<load-on-startup>`, as in `<load-on-startup/>`, the `web.xml` will fail validation against the Servlet 2.4 schema for `web.xml`, causing deployment of the web application to fail.

Backwards compatibility issue. Specifying an empty <load-on-startup> still works with Servlet 2.3 based `web.xml`.

Solution

Specify `<load-on-startup>0</load-on-startup>` when using a Servlet 2.4 based `web.xml` to indicate that servlet load order does not matter.

Unable to compile JSP page on resource constrained servers (6184122)

Description

The JSP page is accessed but fails to compile, and the server log contains the error message "Unable to execute command" with the following stack trace:

```
at org.apache.tools.ant.taskdefs.Execute$Java13CommandLauncher.
exec(Execute.java:655) at org.apache.tools.ant.taskdefs.Execute.
launch(Execute.java:416)
at org.apache.tools.ant.taskdefs.Execute.execute(Execute.java:427)
at org.apache.tools.ant.taskdefs.compilers.DefaultCompilerAdapter.
executeExternalCompile(DefaultCompilerAdapter.java:448)
at org.apache.tools.ant.taskdefs.compilers.JavacExternal.execute
(JavacExternal.java:81)
at org.apache.tools.ant.taskdefs.Javac.compile(Javac.java:842)
at org.apache.tools.ant.taskdefs.Javac.execute(Javac.java:682)
at org.apache.jasper.compiler.Compiler.generateClass(Compiler.java:396)
```

Solution

Set the JSP compilation switch "fork" to "false."

This can be done either of two ways:

- Globally, by setting the fork init parameter of the JspServlet in *domain-dir*`/config/default-web.xml` to false:

```
<servlet> <servlet-name>jsp</servlet-name>
<servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
.... <init-param>
<param-name>fork</param-name> <param-value>false</param-value>
</init-param> .... </servlet>
```

- On a per-web application basis, by setting the fork JSP configuration property in `sun-web.xml` to false:

```
<sun-web-app> <jsp-config> <property name="fork" value="false" />
</jsp-config> </sun-web-app>
```

Either setting will prevent ant from spawning a new process for `javac` compilation.

Web Services

This section describes known web container issues and associated solutions.

Ant task `wsimport` fails with Java EE SDK b33d (using JDK 1.6) with `NoClassDefFoundError` (6527842)

Description

You may encounter a problem when running the JAX—WS tests with the JDK 1.6 included with the Java EE SDK b33d. The tests immediately abort with the following message:

```
[wsimport] Exception in thread "main" java.lang.NoClassDefFoundError: \
com/sun/tools/ws/WsImport
```

This error occurs even though the `webservices-tools.jar` does contain `com/sun/tools/ws/WsImport.class`, `com/sun/tools/ws/ant/WsImport.class`, and `com/sun/tools/ws/ant/WsImport2.class`. Moreover, the same test workspace works without problem using the 1.5.0-10 JDK.

Solution

Copy the `webservices-api.jar` to `$JAVA_HOME/jre/lib/endorsed` before running the JAX-WS tests.

`publish-to-registry` commands fail in IFR EE builds (6602046)

Description

JAXR uses SAAJ to send soap messages to the registry. In the non-IFR case, the SAAJ `impl` classes are under `lib/webservices-rt.jar`. In the IFR case, the SAAJ classes are still under `lib/webservices-rt.jar`. In addition, `saaj-impl.jar` is located in the `/usr/share/lib`

directory. This jar file is picked up by Communications Server and has precedence over classes from `webservices-rt.jar`. This jar file does not have the necessary security permissions to send soap messages to the Web services registry. The packaging should be modified to grant permissions to the jars under `/usr/share/lib` directory or not depend on the `/usr/share/lib` jars.

Solution

Add the following to the `server.policy` file:

```
grant codeBase "file:/usr/share/lib/saaj-impl.jar" {
   permission java.security.AllPermission;
};
```

`wscompile` fails with "package `javax.xml.rpc` does not exist" on JDK6 u4 b3 (6638567)

Description

The `wscompile` ant task fails for JDK 6 Update 4. For each JAX-RPC API class, the following error message is displayed:

`package` *package-name* `does not exist`

Solution

Before running the `wscompile` ant task, ensure that `javaee.jar` is specified in the class path, *not* `j2ee.jar`.